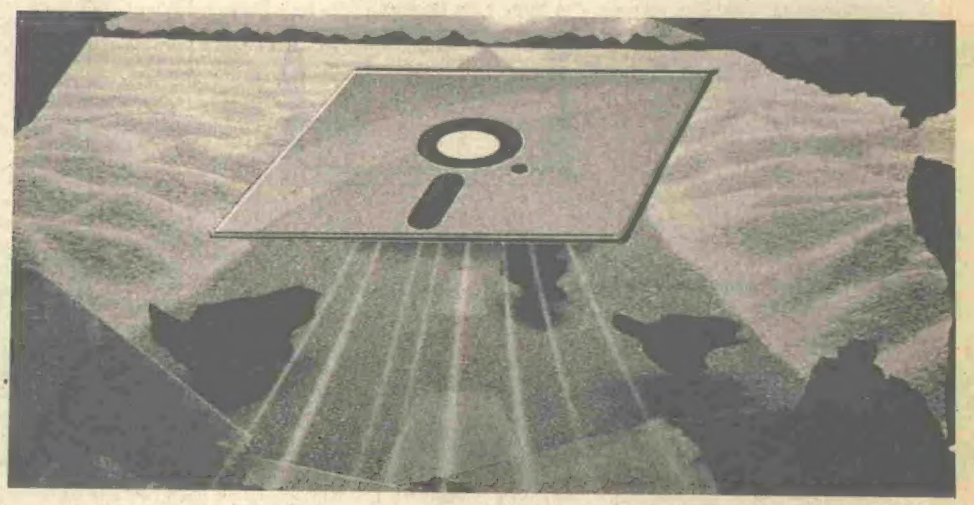
ซอฟต์แวร์
สัมฤทธิ์ แสนธิ

SOFT MAIL

มาสร้างและใช้งานวินโดว์ในโปรแกรมฟอกซ์โปร



ทางเลือกใหม่ที่ดีกว่า เพียง 1 หรือ 2 คำสั่ง ก็สร้างหรือย้ายวินโดว์และข้อมูลตามที่ต้องการ

ผู้อ่านที่เคยเขียนโปรแกรมในดีเบสทรีพลัสมาแล้วและมีแนวโน้มที่จะเปลี่ยนเป็นดีเบสโฟร์ ฟอกซ์เบส หรือฟอกซ์พลัส ซึ่งเป็นโปรแกรมจัดการฐานข้อมูลที่ดีกว่าก่อนที่ผู้อ่านจะเปลี่ยนใจ ผู้เขียนขอแนะนำโปรแกรมจัดการฐานข้อมูลอีกโปรแกรมหนึ่งของบริษัทฟอกซ์ซอฟต์แวร์ คือ โปรแกรมฟอกซ์โปร (FOXPRO) ซึ่งมีข้อดีกว่าโปรแกรมจัดการฐานข้อมูลโดยทั่วไปอยู่มาก ทั้งในด้านความเร็ว จำนวนไฟล์ที่เปิดใช้งานคำสั่งที่มากกว่าซึ่งก็ยังคงสามารถใช้คำสั่งในดีเบสและฟอกซ์เบสได้ทุกคำสั่ง การใช้งานจะเป็นเมนูตอบโต้ตลอดหรือพิมพ์คำสั่งสั่งผ่าน Command Window การเขียนโปรแกรมสามารถเขียนได้ง่ายและสามารถเขียนเป็นแบบโพรซีเดอร์และฟังก์ชันเหมือนกับภาษาระดับสูง (High Level Language) ทำให้สามารถสร้างคำสั่งเก็บไว้ใช้งานได้ นอกจากนี้ฟอกซ์โปรยังสามารถใช้เมาส์ควบคุมการทำงานและควบคุมวินโดว์ได้อย่างน่าทึ่งอีกด้วย

สำหรับบทความบทนี้ ผู้เขียนขอนำเสนอถึงการเขียนโปรแกรมเพื่อสร้างและใช้งานวินโดว์ในฟอกซ์โปร ซึ่งจะมีลักษณะคล้ายกับการเขียนโปรแกรมในดีเบสหรือในฟอกซ์เบส แต่สามารถเขียนได้ง่ายกว่า ทำให้ประหยัดเวลาในการเขียนโปรแกรมมากกว่า บางทีอาจจะทำให้ผู้อ่านตัดสินใจได้ดีขึ้น

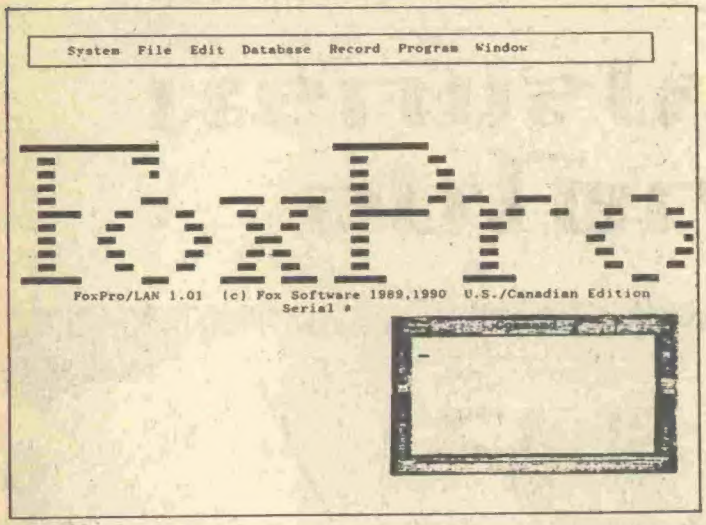
ก่อนเขียนโปรแกรม

เมื่อทำการติดตั้งโปรแกรมฟอกซ์โปรไว้ในฮาร์ดดิสก์แล้ว ควรจะทำการเซตจำนวนไฟล์ที่จะเปิดใช้งานในไฟล์ CONFIG.SYS ให้เหมาะสม แล้วทำการบูตเครื่องใหม่อีกครั้ง เมื่อเรียบร้อยแล้วสั่งพิมพ์

```
C>FOXPROLN <Enter>
```

โปรแกรมฟอกซ์โปรจะถูกเอ็กซีคิวต์ สักครู่หนึ่งจะปรากฏ Screen Window พร้อม Menu และ Command Window ดังรูปที่ 1 ซึ่งก็เปรียบเสมือน Dot Prompt ในดีเบสและฟอกซ์เบสนั่นเอง การใช้งานสามารถเรียกใช้ผ่าน Menu โดยกดปุ่ม F10 และสามารถพิมพ์คำสั่งที่ต้องการใน Command Window เหมือนการใช้งานใน Dot Prompt คำสั่งต่าง ๆ ที่ใช้ในดีเบสและฟอกซ์เบสสามารถใช้งานในฟอกซ์โปรได้ทุกคำสั่ง แต่ต้องมีการ SET COMPATIBLE ในฟอกซ์โปรเสียก่อน โดยถ้าต้องการเอ็กซีคิวต์โปรแกรมที่เขียนขึ้นในฟอกซ์เบสและดีเบสทรีพลัสต้องสั่งเซตโปรแกรมดังนี้

```
        SET COMPATIBLE TO FOXPLUS
หรือ    SET COMPATIBLE OFF
```



รูปที่ 1 แสดง Screen Window และ Command Window

ในทำนองเดียวกันถ้าต้องการเอ็กซีคิวต์โปรแกรมที่เขียนขึ้นในดีเบสโฟร์ต้องสั่งเซตโปรแกรมได้ดังนี้

```
        SET COMPATIBLE TO DB4
หรือ    SET COMPATIBLE ON
```

แต่โดยทั่วไปคำสั่งเบื้องต้นในดีเบสสามารถที่จะนำมาใช้ในฟอกซ์โปรได้เลย เช่น คำสั่ง USE, LIST, MODIFY COMMAND, GO, DISPLAY, DELETE, CREATE, APPEND, BROWSE, DO ฯลฯ

เมนูในฟอกซ์โปรมีทั้งหมด 7 เมนูหลัก แต่ยังมีเมนูอื่น ๆ อีกเช่น Browse Menu เมื่อเรียกใช้คำสั่ง Browse ในแต่ละเมนูหลักจะมีเมนูย่อยให้เลือกใช้งานตามความต้องการอีก สำหรับการใช้งานเมนูหลักมีดังนี้

SYSTEM MENU	ใช้สำหรับแสดงการช่วยเหลือ เครื่องคิดเลข รหัสแอสกี ปฏิทิน ไดอารี เกม เป็นต้น
FILE MENU	ใช้สำหรับสร้างไฟล์ เปิดไฟล์ เซฟโปรแกรม และพรินต์ไฟล์
EDIT MENU	ใช้สำหรับแก้ไขโปรแกรม
DATABASE MENU	ใช้สำหรับแก้ไขข้อมูลในแฟ้มในลักษณะต่าง ๆ
RECORD MENU	ใช้สำหรับเลือกแสดงข้อมูลในเรคอร์ดต่าง ๆ
PROGRAM MENU	ใช้สำหรับการช่วยเขียนโปรแกรม นอกจากนี้ยังสามารถเอ็กซีคิวต์โปรแกรมฟอกซ์วิวและฟอกซ์-ดีอกได้อีกด้วย
WINDOW MENU	ใช้สำหรับเซตวินโดว์ให้แสดงบนหน้าจอในรูปแบบต่าง ๆ จะเห็นว่าในเมนูย่อยจะมี Trace Window และ Debug Window ให้ใช้งานเหมือนการเขียนโปรแกรมในภาษาระดับสูงโดยทั่วไป

นอกจากจะทำการใช้งานฟอกซ์โปรผ่านทางเมนูแล้ว ยังสามารถเรียกใช้งานผ่านทางคีย์บอร์ดได้อีกด้วย ซึ่งมีลักษณะของแต่ละแป้นพิมพ์พอสังเขปดังนี้

F1 ถึง F9	เหมือนในดีเบส
F10	เรียกใช้ Menu
Ctrl+F1	เลือกใช้งาน Window
Ctrl+F2	เรียกใช้งาน Command Window
Ctrl+F7	ย้าย Window
Ctrl+F8	เลือกขนาด Window
Ctrl+F10	Zoom Window
Alt+F	เรียกใช้งาน File Menu
Alt+S	เรียกใช้งาน System Menu
Alt+E	เรียกใช้งาน Edit Menu
Alt+D	เรียกใช้งาน Database Menu
Alt+R	เรียกใช้งาน Record Menu
Alt+P	เรียกใช้งาน Program Menu
Alt+W	เรียกใช้งาน Window Menu
Ctrl+W	Save โปรแกรมหรือข้อมูล
Ctrl+X	สั่ง Cut
Ctrl+A	สั่ง Select All
Ctrl+C	สั่ง Copy
Ctrl+D	สั่ง Do โปรแกรมในแผ่น Diskette
Ctrl+E	สั่ง Replace And Find Again
Ctrl+F	สั่ง Find
Ctrl+G	สั่ง Find Again
Ctrl+K	สั่ง Continue

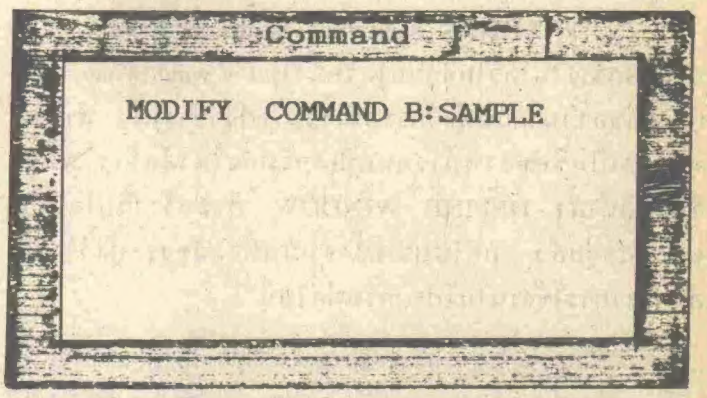
Ctrl+M	สั่ง Resume
Ctrl+O	สั่ง Do โปรแกรมใน Window
Ctrl+Q	
หรือ Esc	ยกเลิก
Ctrl+R	สั่ง Redo
Ctrl+U	สั่ง Undo
Ctrl+V	สั่ง Paste

ความสามารถพิเศษอีกอย่างหนึ่งของโปรแกรมฟอกซ์โปร คือ สามารถใช้งานร่วมกับเมาส์ได้ ซึ่งมีการใช้งานเหมือนโปรแกรมสำเร็จรูปโดยทั่วไป ทำให้สะดวกต่อการใช้งานมาก สำหรับการใช้เมาส์ควบคุมการทำงานของวินโดว์นั้น สามารถทำได้ดังนี้

ACTIVATE WINDOW	→	เลื่อนเคอร์เซอร์ไว้ในวินโดว์ที่ต้องการแล้วกดคีย์เมาส์
CLOSE WINDOW	→	เลื่อนเคอร์เซอร์ไว้มุมกรอบบนด้านซ้ายของวินโดว์แล้วกดคีย์เมาส์
MOVE WINDOW	→	เลื่อนเคอร์เซอร์ไว้บนกรอบด้านบนของวินโดว์แล้วกดคีย์เมาส์ค้างไว้พร้อมทั้งเลื่อนเมาส์
ZOOM WINDOW	→	เลื่อนเคอร์เซอร์ไว้มุมกรอบบนด้านขวาของวินโดว์แล้วกดคีย์เมาส์
SIZE WINDOW	→	เลื่อนเคอร์เซอร์ไว้มุมกรอบล่างด้านขวาของวินโดว์แล้วกดคีย์เมาส์ค้างไว้พร้อมทั้งเลื่อนเมาส์

สำหรับการใช้งาน Command Window จะใช้งานเหมือนใน Dot Prompt ของดีเบส เช่น ถ้าต้องการจะเขียนโปรแกรม ชื่อ SAMPLE.PRG ให้พิมพ์คำสั่งดังรูปที่ 2

สักครู่จะมีวินโดว์สำหรับเขียนโปรแกรม SAMPLE.PRG ให้ ถ้าต้องการให้วินโดว์ขยายใหญ่ให้เลือกกดคีย์ CTRL+F10 หรือ CTRL+F8 เมื่อเขียนโปรแกรมเสร็จเรียบร้อยแล้วต้องการจะเอ็กซีคิวต์โปรแกรม ให้กดคีย์ (TRL+O หรือ ถ้าต้องการให้โปรแกรมทำงานทีละขั้นตอนให้สั่ง SET STEP ON และเช็คการทำงานของโปรแกรมที่ Trace Window



รูปที่ 2 โปรแกรมคำสั่ง SAMPLE.PRG

คำสั่งเกี่ยวกับวินโดว์

คำสั่งที่ใช้งานเกี่ยวกับวินโดว์มีทั้งหมด 12 คำสั่ง ใช้สำหรับสร้าง แสดง ซ่อน ย้าย ลบวินโดว์ ตลอดจนประเภทของวินโดว์ activate และ decativate ซึ่งมีรายละเอียดของคำสั่งได้ดังนี้

1. คำสั่ง DEFINE WINDOW <window name1> FROM <row1>,<col1> TO <row2>,<col2> [IN [WINDOW] <window name2> :SCREEN] [TITLE <expC>] [DOUBLE:PANEL:NONE:SYSTEM:<border string>] [CLOSE:NOCLOSE] [FLOAT:NOFLOAT] [GROW:NOGROW] [SHADOW] [ZOOM:NOZOOM] [COLOR [<standard>], [<enhanced], [<border>]: COLOR SCHEME <expN>]

คำสั่งนี้ค่อนข้างยาวใช้สำหรับสร้างวินโดว์และแอตทริบิวต์ของวินโดว์ ผลของการเอ็กซีคิวต์คำสั่งจะยังไม่แสดงวินโดว์บนหน้าจอ แต่จะเก็บไว้ในหน่วยความจำก่อน ต้องใช้คำสั่ง SHOW WINDOW หรือ ACTIVATE WINDOW จึงจะสามารถแสดงวินโดว์ได้

<window name1> เป็นชื่อของวินโดว์ที่ถูกสร้างไว้ในหน่วยความจำ ประกอบด้วยตัวอักษรไม่เกิน 10 ตัว ขนาดของวินโดว์ถูกกำหนดด้วย <row1>,<col1> ซึ่งเป็นมุมกรอบบนด้านซ้ายของวินโดว์และ <row2>,<col2> เป็นมุมกรอบล่าง ด้านขวาของวินโดว์ ดังตัวอย่างการใช้งานเช่น

```
DEFINE WINDOW SAMPLE FROM 5,5 TO 10,10
```

IN [WINDOW] <window name2>:SCREEN ใช้สำหรับสร้างวินโดว์ในอีกวินโดว์หนึ่งโดย <window name2> เป็นชื่อของวินโดว์ที่ต้องการสร้างวินโดว์ไว้ข้างใน สำหรับหน้าจอเป็นชื่อของหน้าจอมอนิเตอร์ที่ยังไม่ได้สร้างวินโดว์ ซึ่งปกติคำสั่ง DEFINE WINDOW ก็จะสร้างวินโดว์บนหน้าจออยู่แล้ว แต่ในบางครั้งจำเป็นที่จะต้องระบุไว้ ดังตัวอย่างการใช้งานในโปรแกรมต่อไปนี้

```
DEFINE WINDOW SAMPLE0 FROM 5,5 TO 20,20
DEFINE WINDOW SAMPLE1 FROM 2,3 TO 10,10
IN WINDOW SAMPLE0
DEFINE WINDOW SAMPLE1 FROM 2,3 TO 10,10
IN WINDOW SCREEN
```

TITLE <expC> เป็นออปชันสำหรับแสดงข้อความ (<expC>) ที่ด้านบนของกรอบวินโดว์ดังตัวอย่างการใช้งานนี้

```
DEFINE WINDOW SAMPLE FROM 5,20 TO 10,35
TITLE "COMPUTER"
```

DOUBLE:PANEL:NONE:SYSTEM:<border string> เป็นออปชันสำหรับเลือกชนิดของกรอบวินโดว์เลือกใช้ได้ 1 ออปชันเท่านั้น ถ้าไม่ระบุกรอบของวินโดว์จะเป็นเส้นตรงหรือตามคำสั่ง SET BORDER ที่สั่งเซตไว้

รายละเอียดของกรอบวินโดว์แต่ละชนิดมีดังนี้

DOUBLE → เส้นคู่
PANEL → แถบทึบ
NONE → ไม่มีเส้นกรอบ
SYSTEM → จะแสดงสัญลักษณ์ตำแหน่งการใช้เมาส์ไว้บนกรอบของวินโดว์ สำหรับ Option CLOSE, FLOAT, GROW และ ZOOM ส่วนกรอบวินโดว์นั้นจะเป็นเส้นแถบทึบ
<border string> → คือ ตัวอักษร 1 ตัวอักษรที่ต้องการนำมาทำเป็นกรอบวินโดว์ เช่น "A" ดังตัวอย่างการใช้งานในโปรแกรมรูปที่ 3

```
DEFINE WINDOW S1 FROM 5,5 TO 10,10 DOUBLE
DEFINE WINDOW S1 FROM 7,5 TO 12,10 PANEL
DEFINE WINDOW S1 FROM 9,5 TO 14,10 "A"
```

รูปที่ 3 โปรแกรมคำสั่งการนำตัวอักษรมาทำเป็นกรอบวินโดว์

CLOSE:NOCLOSE เมื่อเลือก Close Option ในคำสั่ง DEFINE WINDOW วินโดว์ที่ถูกสร้างขึ้นจะสามารถปิดได้จากการเรียกใช้งาน Close Menu ในเมนูหลักหรือใช้เมาส์ ส่วน Noclose Option จะตรงกันข้ามกับ Close Option ดังตัวอย่างการใช้งานได้ดังนี้

```
DEFINE WINDOW S1 FROM 5,5 TO 10,10 SYSTEM CLOSE
```

FLOAT:NOFLOAT เมื่อเลือก Float Option ในคำสั่ง DEFINE WINDOW วินโดว์ที่ถูกสร้างขึ้นจะสามารถย้ายที่ได้จากการเรียกใช้งาน Move Menu ในเมนูหลักวินโดว์ หรือกดคีย์ CTRL+F7 หรือใช้เมาส์ ส่วน Nofloat Option จะตรงกันข้ามกับ Float Option ดังตัวอย่างการใช้งานได้ดังนี้

```
DEFINE WINDOW S1 FROM 5,5 TO 10,10
SYSTEM CLOSE FLOAT
```

GROW:NOGROW เมื่อเลือก Grow Option ในคำสั่ง DEFINE WINDOW วินโดว์ที่ถูกสร้างขึ้นจะสามารถเลือกขนาดได้จากการเรียกใช้งาน Size Menu ในเมนูหลักวินโดว์ หรือกดคีย์ CTRL+F8 หรือใช้เมาส์ ส่วน Nogrow Option จะตรงข้ามกับ Grow Option ดังตัวอย่างการใช้งานได้ดังนี้

```
DEFINE WINDOW S1 FROM 5,5 TO 10,10
SYSTEM CLOSE FLOAT GROW SHADOW
```

SHADOW เมื่อระบุออปชันนี้ในคำสั่ง DEFINE WINDOW วินโดว์ที่ถูกสร้างขึ้นจะมีเงา ซึ่งปกติจะไม่มี แต่สามารถใช้คำสั่ง SET SHADOW เซตการ On และ Off ได้ ดังตัวอย่างการใช้งานได้ดังนี้

```
DEFINE WINDOW S1 FROM 5,5 TO 10,10
SYSTEM CLOSE FLOAT GROW SHADOW
```

ZOOM:NOZOOM เมื่อเลือก Zoom Option ในคำสั่ง DEFINE WINDOW วินโดว์ที่ถูกสร้างขึ้นจะสามารถ Zoom ได้จากการเรียกใช้งาน Zoom Menu ในเมนูหลักวินโดว์ หรือกดคีย์ CTRL+F10 หรือใช้เมาส์ ส่วน Nozoom Option

จะตรงข้ามกับ Zoom Option ดังตัวอย่างการใช้งานในโปรแกรมนี้

```
DEFINE WINDOW S1 FROM 5,5 TO 10,10
SYSTEM SHADOW ZOOM
```

COLOR [<standard>],[<enhanced>],[<border>] :COLOR SCHEME <expN> ถ้าต้องการให้วินโดว์ที่สร้างขึ้นมีสีสันสวยงามให้ระบุ Color Option หรือ Color Scheme Option ในคำสั่ง DEFINE WINDOW หรืออาจจะใช้คำสั่ง SET COLOR เซตก็ได้ ดังตัวอย่างการใช้งานในโปรแกรมนี้

```
DEFINE WINDOW S1 FROM 5,5 TO 10,10
PANEL TITLE "COMPUTER" SHADOW;
COLOR SCHEME 8
DEFINE WINDOW S2 FROM 8,8 TO 15,15
DOUBLE SHADOW COLOR 7
```

2. คำสั่ง SHOW WINDOW <window name1> [,<window name2>..]:ALL [IN [WINDOW]<window nameN>:SCREEN] [TOP:BOTTOM:SAME] [SAVE]

คำสั่ง SHOW WINDOW เป็นคำสั่งที่ใช้สำหรับแสดงวินโดว์ที่เก็บไว้ในหน่วยความจำ จากคำสั่ง DEFINE WINDOW ที่กล่าวมาแล้วบนหน้าจอ <window name1> [,<window name 2>..] คือชื่อของวินโดว์ที่สร้างขึ้นไว้ในหน่วยความจำ ซึ่งสามารถให้แสดงทีละหลาย ๆ วินโดว์ได้หรือใช้ All Option แทน ดังตัวอย่างการใช้งานในโปรแกรมนี้

```
        DEFINE WINDOW A1 FROM 5,5 TO 10,10
        DOUBLE SHADOW COLOR SCHEME 4
        DEFINE WINDOW A2 FROM 5,20 TO 10,30
        PANEL SHADOW
        SHOW WINDOW A1
        SHOW WINDOW A2
หรือ    SHOW WINDOW A1,A2
หรือ    SHOW WINDOW ALL
```

IN WINDOW <window nameN> : SCREEN เป็นออปชันที่ใช้สำหรับแสดงวินโดว์หนึ่งในอีกวินโดว์หนึ่งหรือจะแสดงบนหน้าจอตามปกติ <window nameN> คือชื่อของวินโดว์ที่ต้องการสร้างวินโดว์ไว้ข้างใน ดังตัวอย่างการใช้งานในโปรแกรมนี้

```
DEFINE WINDOW S1 FROM 5,5 TO 10,10
TITLE "COMPUTER" PANEL SHADOW
DEFINE WINDOW S2 FROM 7,7 TO 15,15
"T" SHADOW COLOR SCHEME 8
SHOW WINDOW S1
SHOW WINDOW S2 IN WINDOW S1
SHOW WINDOW S2 IN WINDOW SCREEN
```

TOP : BOTTOM : SAME เป็นออปชันที่ใช้สำหรับแสดงวินโดว์ไว้ ณ ลำดับต่าง ๆ บนหน้าจอ โดย Top Option หมายถึงด้านบนสุด Bottom Option หมายถึงด้านล่างสุดและ Same Option หมายถึงที่เดิม ปกติการแสดงวินโดว์จากคำสั่ง SHOW WINDOW จะแสดงไว้ด้านบนของวินโดว์ที่แสดงไว้ก่อนเสมอดังตัวอย่างการใช้งานในรูปที่ 4

```
DEFINE WINDOW S1 FROM 5,5 TO 20,20 DOUBLE
DEFINE WINDOW S2 FROM 6,10 TO 8,25 PANEL
DEFINE WINDOW S3 FROM 8,15 TO 22,30 "T"
SHOW WINDOW ALL
WAIT
SHOW WINDOW S2 TOP
WAIT
SHOW WINDOW S2 BOTTOM
```

รูปที่ 4 โปรแกรมคำสั่งการเลือกแสดงวินโดว์ตามลำดับต่าง ๆ

SAVE เป็นออปชันที่ใช้สำหรับแสดงวินโดว์ไว้บนหน้าจอ ถึงแม้จะมีการใช้คำสั่ง DEACTIVATE WINDOW หรือ HIDE WINDOW วินโดว์ที่แสดงไว้จะยังคงอยู่เหมือนเดิม ดังตัวอย่างโปรแกรมนี้

```
DEFINE WINDOW SAMPLE FROM 5,5 TO 10,10
SHADOW COLOR SCHEME 7
SHOW WINDOW SAMPLE BOTTOM SAVE
HIDE WINDOW SAMPLE
```

3. คำสั่ง HIDE WINDOW <window name1> [,<window name 2>..] : ALL IN WINDOW [<window nameN> : SCREEN] [SAVE]

คำสั่ง HIDE WINDOW เป็นคำสั่งที่ใช้สำหรับซ่อนวินโดว์ หรือลบวินโดว์ออกจากหน้าจอหรือวินโดว์ด้วยกัน แต่ยังคงเก็บวินโดว์นั้น ๆ ไว้ในหน่วยความจำ เมื่อเรียกใช้คำสั่ง SHOW WINDOW หรือคำสั่ง ACTIVATE WINDOW จะแสดงวินโดว์นั้น ๆ บนหน้าจอเหมือนเดิม สำหรับออปชันต่าง ๆ ในคำสั่งที่หน้าที่เหมือนในคำสั่ง SHOW WINDOW ดังตัวอย่างการใช้งานในรูปที่ 5

```
        DEFINE WINDOW A1 FROM 1,1 TO 10,10
        DEFINE WINDOW A2 FROM 2,2 TO 5,5
        IN WINDOW A1 DOUBLE
        SHOW WINDOW ALL
        HIDE WINDOW A2 IN WINDOW A1
        HIDE WINDOW A1
หรือ    HIDE WINDOW A2,A1
หรือ    HIDE WINDOW ALL
```

รูปที่ 5 โปรแกรมคำสั่ง HIDE WINDOW

4. คำสั่ง ACTIVATE WINDOW [<window name1> [,<window name2>..]] : ALL [IN [WINDOW] <window nameN : SCREEN] [BOTTOM : TOP : SAME] [NOSHOW]

คำสั่ง ACTIVATE WINDOW เป็นคำสั่งที่ใช้สำหรับแสดงวินโดว์ที่เก็บไว้ในหน่วยความจำเหมือนคำสั่ง SHOW WINDOW และใช้สำหรับ Active Window ทำให้สามารถเขียนข้อความลงในวินโดว์ที่ต้องการได้ <window name1> <,window name2>..] คือชื่อของวินโดว์ที่ต้องการให้ Active จะเห็นว่าสามารถให้ Active ได้ทีละหลาย ๆ วินโดว์ หรือใช้ All Option แทน ส่วนออปชันอื่น ๆ นั้นมีหน้าที่เหมือนในคำสั่ง SHOW WINDOW นอกจาก Noshow Option ถ้าระบุลงในคำสั่งจะทำให้วินโดว์ที่สั่ง Active ไม่แสดงบนหน้าจอหรือวินโดว์ด้วยกัน ดังตัวอย่างการใช้งานในโปรแกรมนี้

```
DEFINE WINDOW S1 FROM 5,5 TO 10,10
DOUBLE SHADOW COLOR SCHEME 8
DEFINE WINDOW S2 FROM 7,2 TO 13,79 SHADOW
ACTIVATE WINDOW S1
?"DEMO ACTIVATE WINDOW"
WAIT
ACTIVATE WINDOW S2
WAIT
```

5. คำสั่ง DEACTIVATE WINDOW <window name1> [, <window name2>..] : ALL

คำสั่ง DEACTIVATE WINDOW จะตรงข้ามกับคำสั่ง ACTIVATE WINDOW ใช้สำหรับลบวินโดว์ออกจากหน้าจอหรือในวินโดว์ด้วยกัน (แต่ยังคงเก็บวินโดว์ไว้ในหน่วยความจำสามารถเรียกมาแสดงเหมือนเดิมได้อีกด้วยคำสั่ง SHOW WINDOW หรือ ACTIVATE WINDOW) และยกเลิกคำสั่ง ACTIVATE WINDOW <window name1> [,<window name2>..] คือชื่อของวินโดว์ที่ต้องการยกเลิกการ Active จะเห็นว่าสามารถยกเลิกได้ที่ละหลาย ๆ วินโดว์ หรือจะใช้ All Option แทน ดังตัวอย่างการใช้งานในโปรแกรมนี้

```
DEFINE WINDOW SAMPLE FROM 5,5 TO 10,10
TITLE "COMPUTER" PANEL
ACTIVATE WINDOW SAMPLE
?"DEMO DEACTIVATE WINDOW"
WAIT
DEACTIVATE WINDOW SAMPLE
```

6. คำสั่ง MOVE WINDOW <window name> <row>, <col> : BY <expN1>, <expN2>

คำสั่ง MOVE WINDOW ใช้สำหรับย้ายตำแหน่งของวินโดว์ให้แสดงในตำแหน่งอื่น ๆ ของหน้าจอโดย

<window name> คือชื่อของวินโดว์ที่ต้องการย้าย

<row>, <col> คือตำแหน่งใหม่ที่ต้องการให้วินโดว์แสดง

<expN1>, <expN2> คือการอ้างตำแหน่งที่จะให้วินโดว์ย้ายไปแบบ Relative โดย expN1 จะเป็นค่าเพิ่มของตำแหน่ง Row และ expN2 จะเป็นค่าเพิ่มของตำแหน่งคอลัมน์ ดังตัวอย่างการใช้งานในโปรแกรมนี้

```
DEFINE WINDOW SAMPLE FROM 5,5 TO 10,10
TITLE "COMPUTER" DOUBLE
SHOW WINDOW SAMPLE
WAIT
MOVE WINDOW SAMPLE TO 10,20
WAIT
MOVE WINDOW SAMPLE BY -5,30
```

7. คำสั่ง SAVE WINDOW <window list> : ALL TO <file> : TO MEMO <memo field>

คำสั่ง SAVE WINDOW ใช้สำหรับเก็บวินโดว์และรายละเอียดของวินโดว์ไว้ใน Magnetic Disk (ได้ไฟล์ มีนามสกุลเป็น.WIN) หรือเก็บไว้ใน Memo Field ของแฟ้มข้อมูลที่เปิดใช้งาน โดยจะใช้คำสั่ง RESTORE WINDOW เรียกกลับคืนมาจาก Magnetic Disk หรือ Memo Field อีกครั้งหนึ่ง

<window list> คือชื่อของวินโดว์ที่ต้องการเก็บไว้ ซึ่งสามารถเก็บได้ทีละหลาย ๆ วินโดว์ได้ โดยระบุชื่อวินโดว์ใน <window list> ทั้งหมดหรือใช้ All Option แทน

<File> คือชื่อไฟล์ที่จะเก็บไว้ใน Magnetic Disk ประกอบด้วยตัวอักษรไม่เกิน 8 ตัวอักษรดังตัวอย่างการใช้งานในโปรแกรมนี้

```
        DEFINE WINDOW S1 FROM 5,5 TO 10,10
        DEFINE WINDOW S2 FROM 10,10 TO 20,20
        SAVE WINDOW S1,S2 TO TEMP
หรือ    SAVE WINDOW ALL TO TEMP
```

<memo field> คือชื่อของ Memo Field ของไฟล์ที่เปิดใช้งาน ซึ่งจะเป็นฟิลด์ที่ใช้เก็บวินโดว์พร้อมกับรายละเอียด ตัวอย่างการใช้งานเช่น ต้องการเก็บวินโดว์และรายละเอียดไว้ใน Memo Field ชื่อ A1 ในไฟล์ CUSTOMER.DBF ดังตัวอย่างการใช้งานในโปรแกรมนี้

```
DEFINE WINDOW OUTPUT FROM 5,5 TO 10,10
DOUBLE FLOAT GROW SHADOW
USE CUSTOMER
SAVE WINDOW OUTPUT TO MEMO A1
USE
```

8. คำสั่ง RESTORE WINDOW <window list> : ALL FROM <file> : FROM MEMO <memo field>

คำสั่ง RESTORE WINDOW ใช้สำหรับเรียกโหลดวินโดว์ พร้อมทั้งรายละเอียดจากไฟล์ใน Magnetic Disk (ไฟล์ที่มีนามสกุล .WIN) หรือ Memo Field ในแฟ้มข้อมูลมาแสดงบนหน้าจอเหมือนเดิม

<window List> คือชื่อของวินโดว์ที่ถูกเก็บไว้ (จากคำสั่ง SAVE WINDOW)

<file> คือชื่อของไฟล์ที่เก็บวินโดว์ไว้ (จากคำสั่ง SAVE WINDOW)

<memo field> คือชื่อของ Memo Field ของแฟ้มข้อมูลที่เปิดอยู่ (จากคำสั่ง SAVE WINDOW)

ดังตัวอย่างการใช้งานในโปรแกรมนี้

```
DEFINE WINDOW A1 FROM 5,5 TO 10,10
TITLE "COMPUTER" SHADOW
ACTIVATE WINDOW A1
?"DEMO RESTORE WINDOW"
WAIT
SAVE WINDOW A1 TO OUTPUT
CLEAR WINDOW
RESTORE WINDOW A1 FROM OUTPUT
```

9. คำสั่ง CLEAR WINDOWS และคำสั่ง RELEASE WINDOWS <window list>

ทั้ง 2 คำสั่งนี้ ใช้สำหรับลบวินโดว์ออกจากหน่วยความจำและหน้าจอ ทำให้ไม่สามารถใช้คำสั่ง SHOW WINDOW และ ACTIVATE WINDOW แสดงวินโดว์ได้อีก สำหรับคำสั่ง CLEAR WINDOWS จะลบทุกวินโดว์ แต่คำสั่ง RELEASE WINDOW จะลบเฉพาะวินโดว์ที่ระบุชื่อใน <window list> ดังตัวอย่างการใช้งานในโปรแกรมนี้

```
        DEFINE WINDOW A1 FROM 5,5 TO 10,10
        DEFINE WINDOW A2 FROM 15,15 TO 20,30
        SHOW WINDOW ALL
        RELEASE WINDOW A1,A2
หรือ    CLEAR WINDOW
```

10. คำสั่ง SET BORDER TO [SINGLE : DOUBLE : PANEL : NONE : <border string1>] [,<border string2>, ..8]

คำสั่งนี้จะใช้สำหรับเลือกชนิดของกรอบ WINDOW, BOXES, MENU และ POPUP ปกติชนิดของกรอบจะเป็นเส้นตรง (Single) หากต้องการให้ชนิดของกรอบเปลี่ยนไปสามารถเลือกได้ 5 แบบ คือ เส้นตรง เส้นคู่ เส้นแถบทึบ ไม่มีเส้นกรอบ ตัวอักษร 1 ตัวที่ต้องการนำมาทำเป็นกรอบ

สำหรับการเลือกตัวอักษร 1 ตัวที่ต้องการทำกรอบ สามารถเลือกใช้ได้ทั้งตัวอักษรตัวเดียวและหลายตัวอักษร เริ่มตั้งแต่ <border string1> ถึง <border string8> ซึ่งใช้เป็นส่วนต่าง ๆ ของกรอบดังนี้ ตัวอักษรที่ 1=> ด้านบน,

2=> ด้านล่าง, 3=> ด้านข้างซ้าย, 4=> ด้านข้างขวา, 5=> มุมซ้ายบน, 6=> มุมขวาบน, 7=> มุมซ้ายล่าง และ 8=> มุมขวาล่าง หากเลือกใช้ตัวอักษรเพียงตัวเดียว ชนิดของกรอบจะเป็นตัวอักษรที่ระบุไว้ในการเลือกตัวอักษร ดังตัวอย่างการเรียกใช้งานในรูปที่ 6

```
SET BORDER TO PANEL
DEFINE WINDOW S1 FROM 5,5 TO 10,10
SHOW WINDOW S1
SET BORDER TO "=","=","|","|","+",
"+","+","+"
DEFINE WINDOW S2 FROM 10,5 TO 15,10
SHOW WINDOW S2
```

รูปที่ 6 โปรแกรมคำสั่งการเลือกชนิดของกรอบ

11. คำสั่ง SET WINDOW OF MEMO TO [<window name>]

คำสั่งนี้เป็นคำสั่งใช้งานวินโดว์ใช้สำหรับเซตข้อมูลใน Memo Field ให้แสดงในวินโดว์ที่ผู้ใช้สร้างขึ้น โดย <window name> คือชื่อของวินโดว์ที่สร้างขึ้นจากคำสั่ง DEFINE WINDOW เมื่อคำสั่งนี้ถูกเอ็กซีคิวต์ จะทำให้สามารถใช้คำสั่ง @ ...SAY/GET, APPEND, BROWSE, CHANGE, EDIT หรือ MODIFY MEMO ในการแก้ไขข้อมูลใน Memo Field ในวินโดว์ที่เซตไว้ได้ สำหรับคำสั่ง SET WINDOW OF MEMO TO จะใช้สำหรับยกเลิกการใช้วินโดว์ให้กลับมาใช้วินโดว์ของระบบ ตัวอย่างการใช้งานเช่น ถ้าต้องการให้ข้อมูลของ Memo Field ชื่อ A1 ของแฟ้มข้อมูล CUSTOMER.DBF แสดงในวินโดว์ที่สร้างขึ้นสามารถเขียนโปรแกรมได้ง่าย ๆ ได้ดังโปรแกรมนี้

```
DEFINE WINDOW S1 FROM 5,5 TO 15,15
TITLE "MEMO" DOUBLE SHADOW
SET WINDOW OF MEMO TO S1
APPEND
SET WINDOW OF MEMO TO
APPEND
```

12. คำสั่ง SET SHADOW ON : OFF

ปกติวินโดว์และ Menu Popups ที่ถูกสร้างนั้นจะไม่มีเงา ถ้าสั่งสร้างแล้วไม่ระบุ Shadow Option ดังนั้นถ้าต้องการให้วินโดว์และ Menu Popups มีเงาเกิดขึ้น สามารถใช้คำสั่ง SET SHADOW บังคับได้ โดย On Option สั่งให้มีเงา และ Off Option จะสั่งให้ไม่มีเงา ดังตัวอย่างการใช้งานในโปรแกรมคำสั่งนี้

```
DEFINE WINDOW A1 FROM 5,5 TO 10,10
DOUBLE COLOR SCHEME 8
SET SHOW ON
SHOW WINDOW A1
SET SHADOW OFF
MOV WINDOW A1 BY 0,20
```

การใช้งานวินโดว์

เมื่อผู้อ่านได้ทดลองใช้ฟอกซ์โปรดูจะเห็นว่าโปรแกรมฟอกซ์โปรได้นำเอาวินโดว์มาใช้ประโยชน์อย่างมากมายเช่น Command Window, Menu, Error Window, Help, Edit, Debug Window, Trace Window และอื่น ๆ อีก นอกจากนี้ยังมีคำสั่งที่นำเอาวินโดว์ไปใช้งานในการเขียนโปรแกรมอีกมากเช่นคำสั่ง BROWSE, CHANGE, EDIT, MODIFY MEMO, CREATE REPORT, MODIFY REPORT, MODIFY LABEL, FILER, CREATE LABEL, HELP ฯลฯ ซึ่งผู้เขียนจะขอเสนอในรายละเอียดในโอกาสต่อไป

ตัวอย่างโปรแกรม DEMO.PRG เป็นโปรแกรมแสดงการใช้งานวินโดว์อย่างง่าย ๆ ขอให้ผู้อ่านทดลองเช็คการทำงานดู จะทำให้เข้าใจในการใช้คำสั่งดียิ่งขึ้น

สรุปส่งท้าย

โปรแกรมสำเร็จรูปในปัจจุบันโดยส่วนใหญ่จะเห็นว่านำเอาวินโดว์มาใช้กันอย่างกว้างขวางในยุคแรก ๆ ของการเขียนโปรแกรมเพื่อที่จะสร้างและเก็บข้อมูลในวินโดว์นั้น ทำได้ค่อนข้างยากโปรแกรมที่เขียนขึ้นจะมีความยาวมากทำงานได้ช้า ยิ่งเป็นโปรแกรมสำเร็จรูปเช่น ดีเบส จะทำการเขียนโปรแกรมได้ยากมาก เนื่องจากไม่เหมือนภาษาของคอมพิวเตอร์เองจึงมีขีดจำกัดมาก แต่ปัจจุบันนี้คำสั่งในโปรแกรมฟอกซ์โปรเพียง 1 หรือ 2 คำสั่งก็สามารถสร้างหรือย้ายวินโดว์และข้อมูลในวินโดว์ให้แสดงตามตำแหน่งต่าง ๆ บนหน้าจอได้อย่างรวดเร็ว

จึงได้มีการนำเอาวินโดว์ไปใช้ประโยชน์อย่างมากมายทำให้การใช้งานคล่องตัวขึ้น ในอนาคตข้างหน้านี้มีแนวโน้มว่าฟอกซ์โปรจะมาแทนที่ดีเบสอย่างแน่นอน เมื่อผู้อ่านอ่านมาถึงจุดนี้แล้วคงจะเกิดการตัดสินใจแล้วซินะครับว่าจะเลือกโปรแกรมจัดการฐานข้อมูลโปรแกรมไหนดีฉบับหน้าพบกันตอนใหม่ครับ

โปรแกรม DEMO.PRG

```
set talk off
set cursor off
clear
release all
clear window
dir *.*
do showwin
wait "" timeout 10

do msg2user
release window msg2user
do releasewin
set talk on
set cursor on
*******************************
procedure showwin
  r=1
  c=1
  for i=1 to 30
    wait "" timeout 1
    sd="a"+alltrim(str(i))
    define window &sd from r,c to r+2,c+12 none shadow color scheme 8
    activate window &sd
    @ 1,1 say 'DEMO WINDOW'
    c=c+16
    if c>=70
       r=r+4
       c=1
    endif
  endfor
  return
*********************************************************
procedure msg2user
   define window msg2user from 9,0 to 14,79  color scheme 7
   activate window msg2user
   clear
   text
     THINK-YOU FOR USING DEMO FROM WRITER OF FOXPRO
 FROM FOX SOFTWARE OF PERRYSBURG , OH. IF YOU HAVE ANY QUESTIONSOR
 IDES CONCERNING DEMO , PLEASE CALL TEL 3239009.
   ENDTEXT
   wait " press any key to continue....." timeout 10
   return
*************************************************
procedure releasewin
   for i=30 to 1 step -1
     wait "" timeout 1
     sd="a"+alltrim(str(i))
     release window &sd
   endfor
   return
```